Write three Python functions as specified below. Paste the text for all three functions together into the submission window. Your function will be called automatically with various inputs and should return values as specified. Do not write commands to read any input or print any output.

1. Write a function `intreverse(n)` that takes as input a positive integer `n` and returns the integer obtained by reversing the digits in `n`.

Here are some examples of how your function should work.

```python
>>> intreverse(783)
387
>>> intreverse(242789)
987242
>>> intreverse(3)
```

Due on 2019-08-22, 23:59 IST
2. Write a function `matched(s)` that takes as input a string `s` and checks if the brackets "(" and ")" in `s` are matched: that is, every "(" has a matching ")" after it and every ")" has a matching "(" before it. Your function should ignore all other symbols that appear in `s`. Your function should return `True` if `s` has matched brackets and `False` if it does not.

Here are some examples to show how your function should work.

```python
>>> matched("zb%78")
True
>>> matched("(7){a}"
False
>>> matched("a)*{?"
False
>>> matched("((jkl)78(A)&l{8(dd(FJI:),):?):})")
True
```

3. Write a function `sumprimes(l)` that takes as input a list of integers `l` and returns the sum of all the prime numbers in `l`.

Here are some examples to show how your function should work.

```python
>>> sumprimes([3,3,1,13])
19
>>> sumprimes([2,4,6,9,11])
13
>>> sumprimes([-3,1,6])
0
```

**Sample Test Cases**

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case 1</td>
<td><code>intreverse(31511)</code></td>
<td>11513</td>
</tr>
<tr>
<td>Test Case 2</td>
<td><code>intreverse(4)</code></td>
<td>4</td>
</tr>
<tr>
<td>Test Case 3</td>
<td><code>intreverse(15135324234235)</code></td>
<td>5324324235 3151</td>
</tr>
<tr>
<td>Test Case 4</td>
<td><code>matched(&quot;a3qw3;4w3(aasdgds) ((agadsg dsgag)agaga&quot;)&quot;)</code></td>
<td>True</td>
</tr>
</tbody>
</table>
| Test Case 5 | `matched("{ag(Gaga(agag)Gaga)GG)a}33)
cc(""` | False               |
| Test Case 6 | `matched("{adsgdsg(agaga)a")` | False               |
| Test Case 7 | `matched("15ababa.agaga[][["]")` | True                |
The due date for submitting this assignment has passed.
As per our records you have not submitted this assignment.

Sample solutions (Provided by instructor)

```python
def intreverse(n):
    ans = 0
    while n > 0:
        (d, n) = (n % 10, n // 10)
        ans = 10 * ans + d
    return (ans)
def matched(s):
    nested = 0
    for i in range(0, len(s)):
        if s[i] == "(":
            nested += 1
        elif s[i] == ")":
            nested -= 1
        if nested < 0:
            return (False)
    return (nested == 0)
def factors(n):
    factorlist = []
    for i in range(1, n+1):
        if n % i == 0:
            factorlist = factorlist + [i]
    return (factorlist)
def isprime(n):
    return (factors(n) == [1, n])
def sumprimes(l):
    sum = 0
    for i in range(0, len(l)):
        if isprime(l[i]):
            sum = sum + l[i]
```

<table>
<thead>
<tr>
<th>Test Case</th>
<th>sumprimes([101, 93, 97, 44])</th>
<th>198</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>sumprimes([1001, 393, 743, 59])</td>
<td>802</td>
</tr>
<tr>
<td>Test Case</td>
<td>sumprimes([11, 11, 11, 13, 11, -11])</td>
<td>57</td>
</tr>
<tr>
<td>Test Case</td>
<td>intreverse(368)</td>
<td>863</td>
</tr>
<tr>
<td>Test Case</td>
<td>intreverse(798798)</td>
<td>897897</td>
</tr>
<tr>
<td>Test Case</td>
<td>intreverse(7)</td>
<td>7</td>
</tr>
<tr>
<td>Test Case</td>
<td>matched(&quot;(7)(a&quot;)</td>
<td>False</td>
</tr>
<tr>
<td>Test Case</td>
<td>matched(&quot;a)*(?&quot;</td>
<td>False</td>
</tr>
<tr>
<td>Test Case</td>
<td>matched(&quot;((jkl)78(A)&amp;l(8(dd(FJ I:)),:))&quot;)&quot;</td>
<td>True</td>
</tr>
<tr>
<td>Test Case</td>
<td>sumprimes([17, 51, 29, 39])</td>
<td>46</td>
</tr>
<tr>
<td>Test Case</td>
<td>sumprimes([-3, -5, 3, 5])</td>
<td>8</td>
</tr>
<tr>
<td>Test Case</td>
<td>sumprimes([4, 6, 15, 27])</td>
<td>0</td>
</tr>
</tbody>
</table>
return(sum)

def tolist(inp):
    inp = "["+inp+""]
    inp = ast.literal_eval(inp)
    return (inp[0],inp[1])

def parse(inp):
    inp = ast.literal_eval(inp)
    return (inp)

fncall = input()
lparen = fncall.find("(")
rparen = fncall.rfind(")")
fname = fncall[lparen]
farg = fncall[lparen+1:rparen]

if fname == "intreverse":
    arg = parse(farg)
    print(intreverse(arg))
elif fname == "matched":
    arg = parse(farg)
    print(matched(arg))
elif fname == "sumprimes":
    arg = parse(farg)
    print(sumprimes(arg))
else:
    print("Function", fname, "unknown")